

ELABORACIÓN DE PROTOTIPO ANDROID CON GESTIÓN DE  
AUTENTICACIÓN Y ROLES PARA ADMINISTRACIÓN DE CONTENIDOS PARA  
CÁTEDRA UNIVERSITARIA CON ÉNFASIS EN LIDERAZGO EN LA  
UNIVERSIDAD TECNOLÓGICA DE PEREIRA.

JHON ANDREY CASTAÑO QUINTERO

UNIVERSIDAD TECNOLÓGICA DE PEREIRA  
FACULTAD DE INGENIERÍAS ELÉCTRICAS, ELECTRÓNICA, FÍSICA Y  
CIENCIAS DE LA COMPUTACIÓN  
PEREIRA  
2021

ELABORACIÓN DE PROTOTIPO ANDROID CON GESTIÓN DE  
AUTENTICACIÓN Y ROLES PARA  
ADMINISTRACIÓN DE CONTENIDOS PARA CÁTEDRA UNIVERSITARIA CON  
ÉNFASIS EN LIDERAZGO EN LA UNIVERSIDAD TECNOLÓGICA DE PEREIRA.

PROYECTO DE GRADO PARA OPTAR AL TÍTULO DE:  
INGENIERO DE SISTEMAS Y COMPUTACIÓN

DIRECTOR DE PROYECTO DE GRADO  
JUAN DE JESUS VELOZA  
INGENIERO ELECTRICO

UNIVERSIDAD TECNOLÓGICA DE PEREIRA  
FACULTAD DE INGENIERIAS ELECTRICA, ELECTRONICA, FISICA Y  
CIENCIAS DE LA COMPUTACION  
PROGRAMA: INGENIERIA DE SISTEMAS Y COMPUTACION  
PEREIRA  
2019  
AGRADECIMIENTOS

“Agradezco a mis abuelos, y a las personas que contribuyeron de manera directa e indirecta con su apoyo.”

## Tabla de contenido

<b>Capítulo I: formulación del problema.</b>	<b>6</b>
<b>1.1 Introducción.</b>	<b>6</b>
<b>1.2 Descripción del problema.</b>	<b>6</b>
<b>1.3 Justificación.</b>	<b>6</b>
<b>1.4 Objetivos.</b>	<b>7</b>
1.4.1 Objetivo general.	7
1.4.2 Objetivos específicos.	7
<b>1.5 Alcance de la propuesta.</b>	<b>7</b>
<b>Capítulo II: Marco teórico</b>	<b>7</b>
<b>2.1 Marco referencial</b>	<b>7</b>
MOODLE:	7
NODE JS:	8
FRAMEWORK:	8
IONIC:	8
JAVASCRIPT:	8
TYPESCRIPT:	8
ANGULAR:	9
APACHE CORDOVA:	9
SHELL:	9
VISUAL STUDIO CODE:	9
PWA:	9
GOOGLE ANALYTICS:	9
FIREBASE:	9
HTML5:	10
CSS:	10
API REST:	10
MICROSERVICIOS:	10
WEB SERVICES:	11
AJAX:	11
XML:	11
JSON:	11
WEB COMPONENTS:	11
UI (USER INTERFACE):	12
GUI:	12
UX (USER EXPERIENCE):	12
NPM:	12
DOCKER:	12
OAUTH2:	12
JSON WEB TOKEN:	13
LAZY LOADING:	13
<b>Capítulo III:</b>	<b>13</b>
<b>Marco metodológico</b>	<b>13</b>
<b>3.1 Metodología para el análisis y diseño general de la solución.</b>	<b>13</b>
Diagrama de casos de uso.	14
Historias de usuario.	15
<b>3.2 Proceso de actualización Ionic v3 Ionic 5.4.16.</b>	<b>16</b>

<b>3.2.1 Estructura del proyecto .....</b>	<b>16</b>
Cambios de RxJS .....	18
Eventos de ciclo de vida. ....	18
Componentes superpuestos. ....	18
Navegación. ....	18
Configuración de entorno de trabajo.....	18
<b>3.2.2 PROCESO DE AUTENTIFICACIÓN EN LA NUEVA VERSION DE IONIC. ....</b>	<b>23</b>
<b><i>Capítulo IV: Aspectos administrativos.....</i></b>	<b>27</b>
<b>4.1 Recursos.....</b>	<b>27</b>
<b>4.2 Cronograma .....</b>	<b>28</b>
<b><i>Trabajo futuro.....</i></b>	<b>28</b>
<b><i>Conclusiones .....</i></b>	<b>29</b>
<b><i>Anexos .....</i></b>	<b>29</b>
Repositorio aplicativo .....	29
Lista Web Services moodle_mobile_app.....	29
<b><i>Bibliografía .....</i></b>	<b>55</b>

## Tabla de ilustraciones

Ilustración 1.Diagrama de caso de uso .....	15
Ilustración 2.Comparativa estructura ionic v3-v4.....	17
Ilustración 3.Sitio nodejs.org/es .....	19
Ilustración 4.Comando node –versión mac .....	20
Ilustración 5.Comando cordova --version mac .....	20
Ilustración 6.Sitio Download VScode.....	21
Ilustración 7.Resultado ionic g page login .....	24
Ilustración 8.Resultado ionic g service .....	24
Ilustración 9.Template login moodle .....	25
Ilustración 10.Vista Login .....	26
Ilustración 11.Imagen Docker proyecto.....	27
Ilustración 12.Recursos desarrollo.....	27
Ilustración 13.Cronograma proyecto.....	28

# **Capítulo I: formulación del problema.**

## **1.1 Introducción.**

En la universidad tecnológica de Pereira a finales del primer semestre del 2018 cerca del 37% de los jóvenes que ingresaron al primer semestre desertaron. Para combatir este indicador nace un proyecto llamado “Cátedra Universitaria con énfasis en liderazgo”. Presentado por el ingeniero y docente Juan de Jesus Veloza, el cual pretende solucionar distintos problemas que experimentan los estudiantes a lo largo de su vida universitaria, con ayuda de las herramientas TIC. En la actualidad varias tesis han contribuido con la realización de este proyecto en forma articulada. El aporte desde el presente proyecto busca continuar con el proceso que se ha llevado, elaborando sistema de roles y autenticación que permita controlar el acceso a los contenidos presentados en el curso virtual.

El uso e instalación de aplicaciones en smartphone se ha visto incrementada en los últimos años, muchas aplicaciones son descargadas desde sus tiendas oficiales en app Store y google play las cuales dominan el mercado, los smartphone se han convertido en una tecnología de comunicación indispensable para nuestras vidas cotidianas.

Este proyecto pretende actualizar el aplicativo desarrollado en un proyecto anterior el cual permite dar mayor funcionalidad y comodidad a los estudiantes de primer semestre.

## **1.2 Descripción del problema.**

Para el año 2018 se ha planteado un proyecto denominado “Cátedra Universitaria con énfasis en liderazgo”, presentado por el ingeniero y docente Juan de Jesus Veloza, el cual busca reducir la deserción estudiantil incentivando el auto aprendizaje con ayuda de la plataforma virtual y las tecnologías móviles. En 2019 un proyecto fue presentado como base para la elaboración de un aplicativo Android llamado: “Desarrollo de aplicación para dispositivos móviles Android enfocado a guiar a los estudiantes recién ingresados a la Universidad Tecnológica de Pereira, como complemento al curso virtual Catedra universitaria con énfasis en liderazgo.” Muchas universidades actualmente están incentivando al auto aprendizaje con plataformas virtuales, generando objetos virtuales de aprendizaje (OVA) de calidad que aumentan la compenetración entre estudiante y campus. Para poder calcular el impacto de estos cursos se necesitan datos que permitan su evaluación, e ir mejorando la calidad conforme a el análisis de estos.

## **1.3 Justificación.**

La implementación de una solución tecnológica no garantiza el éxito de esta, el impacto que cause es un factor determinante, para que un proyecto tecnológico fracase o continúe en vigencia. Constantemente surgen nuevas tecnologías y estándares aplicados, que permiten incrementar el desempeño y experiencia de usuario, los proyectos de software que van enfocados a un uso masivo necesitan adaptarse a las necesidades del usuario. Para conocer como ha impactado nuestra solución tecnológica necesitamos medir y comparar con un estado inicial, estos valores que llamamos métricas y son usados frecuentemente para tener el conocimiento sobre si un proyecto ha sido exitoso. En el proyecto catedra universitaria con énfasis en liderazgo se desarrolló un prototipo inicial de aplicativo móvil que consta de diferentes módulos. Sobre el cual se pretende agregar nuevas funcionalidades con el fin de ponerlo en marcha y poder medir el impacto de este.

## **1.4 Objetivos.**

### **1.4.1 Objetivo general.**

Elaboración de prototipo Android con gestión de autenticación y roles para administración de contenidos para cátedra universitaria con énfasis en liderazgo en la Universidad Tecnológica de Pereira.

### **1.4.2 Objetivos específicos.**

- Construir ingeniería de software para el aplicativo Android.
- Migrar aplicativo de ionic3 a ionic4
- Desarrollar módulo de autenticación en la plataforma Moodle
- Realizar y ejecutar un plan de pruebas.

## **1.5 Alcance de la propuesta.**

El prototipo permite la autenticación en la plataforma Moodle, e inscribirse al curso catedra universitaria con énfasis en liderazgo. Cuenta con una versión pwa (progressive web app) que permite añadir nuevos módulos, de acuerdo a las necesidades que se presenten a medida que avanza el proyecto.

## **Capitulo II: Marco teórico**

### **2.1 Marco referencial.**

#### **MOODLE:**

Moodle (pronunciación AFI) es una herramienta de gestión de aprendizaje, o más concretamente de Learning Content Management, de distribución libre, escrita en PHP. Está concebida para ayudar a los docentes a crear comunidades de

aprendizaje en línea, Moodle es usada en blended learning, educación a distancia, clase invertida y diversos proyectos de e-learning en escuelas, universidades, oficinas y otros sectores. La versión más reciente es la 3.7. (Moodle, s.f.)

### **NODE JS:**

Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación ECMAScript, asíncrono, con I/O de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google. Fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables, como, por ejemplo, servidores web. Fue creado por Ryan Dahl en 2009 y su evolución está apadrinada por la empresa Joyent, que además tiene contratado a Dahl en plantilla. (Node.js, s.f.)

### **FRAMEWORK:**

Entorno de trabajo o marco de trabajo es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar. Un framework para aplicaciones web es un framework diseñado para apoyar el desarrollo de sitios web dinámicos, aplicaciones web y servicios web. Este tipo de frameworks intenta aliviar el exceso de carga asociado con actividades comunes usadas en desarrollos web. Por ejemplo, muchos framework proporcionan bibliotecas para acceder a bases de datos, estructuras para plantillas y gestión de sesiones, y con frecuencia facilitan la reutilización de código. (Framework para aplicaciones web, s.f.)

### **IONIC:**

Ionic Framework es un kit de herramientas de interfaz de usuario de código abierto para crear aplicaciones móviles y de escritorio de alta calidad y de alto rendimiento utilizando tecnologías web (HTML, CSS y JavaScript).  
<https://ionicframework.com/docs/intro>

### **JAVASCRIPT:**

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. (Javascript, s.f.)

### **TYPESCRIPT:**

Programación libre y de código abierto desarrollado y mantenido por Microsoft. Es un superconjunto de JavaScript, que esencialmente añade tipos estáticos y objetos basados en clases (TypeScript, s.f.).



## ANGULAR:

Es un framework de JavaScript de código abierto, mantenido por google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador, en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles (AngularJS, s.f.)

## APACHE CORDOVA:

Es un marco de código abierto que permite a los desarrolladores web centrarse en el desarrollo de apps para teléfonos inteligentes de distintas plataformas, teniendo como base un código genérico con herramientas tales como JavaScript, HTML, CSS, y creando una interfaz de funciones foráneas para embeber una vista Web en el dispositivo móvil. (Apache Cordova, s.f.)

## SHELL:

Es el término usado en informática para referirse a un intérprete de comandos, el cual consiste en la interfaz de usuario tradicional de los sistemas operativos basados en Unix y similares, como GNU/Linux. (Shell de Unix, s.f.).

## VISUAL STUDIO CODE:

Es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto, aunque la descarga oficial está bajo software propietario requiriendo tus datos de uso del programa legalmente (Visual Studio Code, s.f.)

## PWA:

Su traducción literal al español sería: Aplicaciones Web Progresivas, aunque esta traducción no desvela un significado muy concreto ya que, realmente, las Progressive Web Applications o PWA, son la evolución de las páginas web hacia un ecosistema tecnológico más cercano al de una aplicación móvil o de escritorio instalable en el sistema operativo. (Progressive Web Apps, s.f.)

## GOOGLE ANALYTICS:

Es una herramienta de analítica web de la empresa Google. Ofrece información agrupada del tráfico que llega a los sitios web según la audiencia, la adquisición, el comportamiento y las conversiones que se llevan a cabo en el sitio web.

## FIREBASE:

Es una plataforma ubicada en la nube, integrada con Google Cloud Platform, que usa un conjunto de herramientas para la creación y sincronización de proyectos.

## HTML5:

HTML, siglas en inglés de HyperText Markup Language, hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros. Es un estándar a cargo del World Wide Web Consortium o Consorcio WWW, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. Se considera el lenguaje web más importante siendo su invención crucial en la aparición, desarrollo y expansión de la World Wide Web. Es el estándar que se ha impuesto en la visualización de páginas web y es el que todos los navegadores actuales han adoptado. (HTML, s.f.)

## CSS:

En español «Hojas de estilo en cascada», es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML o XHTML; el lenguaje puede ser aplicado a cualquier documento XML, incluyendo XHTML, SVG, XUL, RSS, etcétera. Te puede ayudar a crear tu propio sitio web. Junto con HTML y JavaScript, CSS es una tecnología usada por muchos sitios web para crear páginas visualmente atractivas, interfaces de usuario para aplicaciones web y GUIs para muchas aplicaciones móviles. (Hoja de estilos en cascada, s.f.)

## API REST:

Es un tipo de arquitectura de desarrollo web que se apoya totalmente en el estándar HTTP

Una API de transferencia de estado representacional (REST), o API de RESTful, es una interfaz de programación de aplicaciones (API o API web) creada por el informático Roy Fielding, la cual se ajusta a los límites de la arquitectura REST y permite la interacción con los servicios web de RESTful. (hat, 2021)

## MICROSERVICIOS:

La arquitectura de microservicios es una aproximación para el desarrollo de software que consiste en construir una aplicación como un conjunto de pequeños servicios, los cuales se ejecutan en su propio proceso y se comunican con mecanismos ligeros. Cada servicio se encarga de implementar una funcionalidad completa del negocio. Cada servicio es desplegado de forma independiente y puede estar programado en distintos lenguajes y usar diferentes tecnologías de almacenamiento de datos. (Arquitectura de microservicios, s.f.)

## WEB SERVICES:

Un servicio web es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web. (Servicio web, s.f.)

## AJAX:

Acrónimo de Asynchronous JavaScript And XML, es una técnica de desarrollo web para crear aplicaciones interactivas o RIA. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones. (Ajax, s.f.)

## XML:

Es una recomendación W3C para proporcionar elementos y atributos con nombre único en un archivo XML. Un archivo XML puede contener nombres de elementos o atributos procedentes de más de un vocabulario XML. Si a cada uno de estos vocabularios se le da un espacio de nombres, un ámbito semántico propio, referenciado a una URI donde se listen los términos que incluye, se resuelve la ambigüedad existente entre elementos o atributos que se llamen igual, la homonimia. Los nombres de elementos dentro de cada espacio de nombres deben ser únicos. (Espacio de nombres XML, s.f.)

## JSON:

Es un formato de texto sencillo para el intercambio de datos. Se trata de un subconjunto de la notación literal de objetos de JavaScript, aunque, debido a su amplia adopción como alternativa a XML, se considera un formato independiente del lenguaje (JSON, s.f.).

## WEB COMPONENTS:

Los componentes web son un conjunto de características que actualmente están siendo añadidas por el W3C a las especificaciones HTML y DOM de forma que permite la creación de widgets o componentes reutilizables en documentos y aplicaciones web. La intención de estos componentes es traer la ingeniería basada en componentes a la World Wide Web. El modelo de componentes permite la encapsulación e interoperabilidad de elementos de HTML individuales. (Componentes web, s.f.)

## UI (USER INTERFACE):

Se refiere a la interfaz gráfica de una plataforma o sitio web que es visitada por usuarios.

## GUI:

La interfaz gráfica de usuario, conocida también como GUI, es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz. Su principal uso, consiste en proporcionar un entorno visual sencillo para permitir la comunicación con el sistema operativo de una máquina o computador. (Interfaz gráfica de usuario, s.f.)

## UX (USER EXPERIENCE):

La experiencia de usuario es el conjunto de factores y elementos relativos a la interacción del usuario con un entorno o dispositivo concretos, dando como resultado una percepción positiva o negativa de dicho servicio, producto o dispositivo. Esta depende no solo de los factores relativos al diseño sino de aspectos relativos a las emociones, sentimientos, construcción y transmisión de la marca, confiabilidad del producto, etc. (Experiencia de usuario, s.f.)

## NPM:

npm es el sistema de gestión de paquetes por defecto para Node.js, un entorno de ejecución para JavaScript, bajo Artistic License 2.0. Desde la versión 0.6.3 de Node.js npm es instalado automáticamente con el entorno. npm se ejecuta desde la línea de comandos y maneja las dependencias para una aplicación. Además, permite a los usuarios instalar aplicaciones Node.js que se encuentran en el repositorio. npm está escrito enteramente en JavaScript y fue desarrollado por Isaac Z. Schlueter a raíz de la frustración que experimentó mientras trabajaba con CommonJS y considerando las deficiencias de otros proyectos similares como PHP y Perl. (Npm, s.f.)

## DOCKER:

es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos. Docker utiliza características de aislamiento de recursos del kernel Linux, tales como cgroups y espacios de nombres para permitir que "contenedores" independientes se ejecuten dentro de una sola instancia de Linux, evitando la sobrecarga de iniciar y mantener máquinas virtuales. (Docker, s.f.)

## OAuth2:

Open Authorization es un estándar abierto que permite flujos simples de autorización para sitios web o aplicaciones informáticas. Se trata de un protocolo propuesto por Blaine Cook y Chris Messina, que permite autorización segura de una

API de modo estándar y simple para aplicaciones de escritorio, móviles y web. (OAuth, s.f.)

### **JSON WEB TOKEN:**

Es un estándar abierto basado en JSON propuesto por IETF para la creación de tokens de acceso que permiten la propagación de identidad y privilegios o claims en inglés. Por ejemplo, un servidor podría generar un token indicando que el usuario tiene privilegios de administrador y proporcionarlo a un cliente. El cliente entonces podría utilizar el token para probar que está actuando como un administrador en el cliente o en otro sistema. El token está firmado por la clave del servidor, así que el cliente y el servidor son ambos capaz de verificar que el token es legítimo. Los JSON Web Tokens están diseñados para ser compactos, poder ser enviados en las URLs -URL-safe- y ser utilizados en escenarios de Single Sign-On. Los privilegios de los JSON Web Tokens pueden ser utilizados para propagar la identidad de usuarios como parte del proceso de autenticación entre un proveedor de identidad y un proveedor de servicio, o cualquiera otro tipo de privilegios requeridos por procesos empresariales. (JSON Web Token, s.f.)

### **LAZY LOADING:**

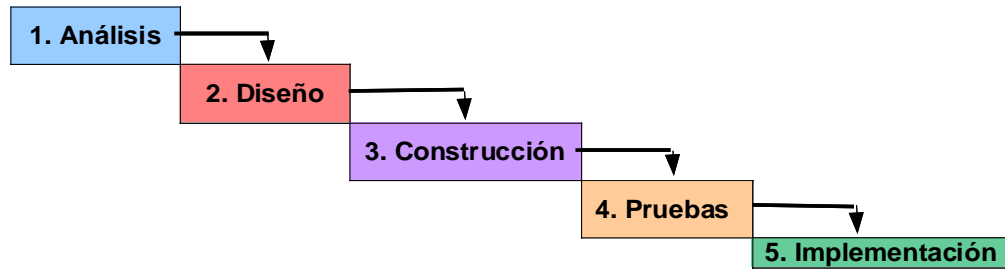
La carga diferida es un patrón de diseño comúnmente usado en la programación informática que consiste en retrasar la carga o inicialización de un objeto hasta el mismo momento de su utilización. Esto contribuye a la eficiencia de los programas, evitando la precarga de objetos que podrían no llegar a utilizarse. El opuesto de la carga diferida es la carga previa, precarga o eager loading. (Lazy loading, s.f.)

## **Capítulo III:**

### **Marco metodológico**

#### **3.1 Metodología para el análisis y diseño general de la solución.**

La metodología en cascada es un modelo lineal de diseño de software que emplea un proceso de diseño secuencial. El desarrollo fluye secuencial mente desde el punto inicial hasta el punto final, con varias etapas diferentes: análisis, diseño, construcción, pruebas, implementación.[3]



Metodología cascada prototipo android  
realizada por: Jhon Andrey Castaño  
Figura 1

Fase de Análisis: Se identifican y registran los requerimientos necesarios para el desarrollo del prototipo, diseño de casos de uso, diseño de diagrama de objetos, relaciones y jerarquía de contenidos, diseño de diagramas de clases

Fase de diseño: Se realizan los diseños de mockups, diagrama de despliegue (modelo de configuración) diseño de diagrama de clases (diseño modelo vista controlador), diseño de modelo relacional para la base de datos.

Fase de construcción: Se programan los casos de uso y se implementa la base de datos.

Fase de pruebas: Se elabora plan de pruebas, se ejecutan pruebas unitarias y pruebas integrales se realiza el informe de resultados de las pruebas.

Fase de implementación: Se migra la base de datos, se instala la aplicación y se redacta el manual de usuario.

9.

Diagrama de casos de uso.

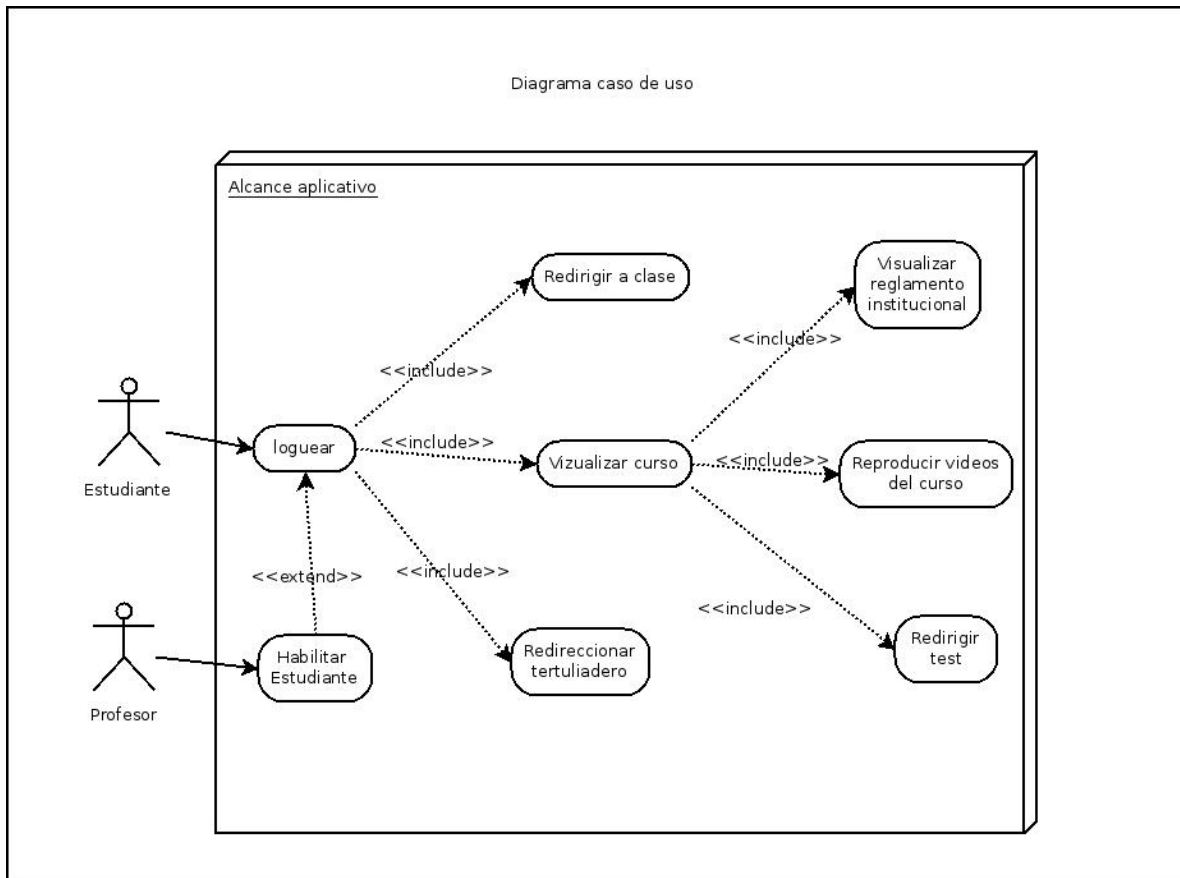


Ilustración 1. Diagrama de caso de uso

## Historias de usuario.

### Prioridad

Alto	Medio	Bajo
3	2	1

ID:	HU1	NOMBRE:	actualización de framework ionic
PRIORIDAD:	3	ITERACION ASIGNADA:	
DESCRIPCION:			
Actualizar versión del framework ionic del prototipo actual a la version ionic 4.			
OBSERVACIONES	Actualizar componentes de acuerdo con los nuevos estándares del framework		
CRITERIO DE ACEPTACION	módulos actualizados para uso en ionic 4.		

ID:	HU2	NOMBRE:	autenticación
-----	-----	---------	---------------

PRIORIDAD:	3	ITERACION ASIGNADA:	
DESCRIPCION:			
El usuario podrá autenticarse en la aplicación para poder ingresar a los módulos			
OBSERVACIONES	El ingreso del administrador del curso podrá acceder a un módulo extra de administración.		
CRITERIO DE ACEPTACION	Ingreso exitoso		

ID:	HU3	NOMBRE:	Sistema pwa
PRIORIDAD:	3	ITERACION ASIGNADA:	
DESCRIPCION:			
la aplicación podrá instalarse desde un explorador que acepte pwa			
OBSERVACIONES			
CRITERIO DE ACEPTACION			

ID:	HU4	NOMBRE:	Módulo de métricas
PRIORIDAD:	3	ITERACION ASIGNADA:	
DESCRIPCION:			
El administrador podrá acceder a un módulo que permita identificar ingreso de usuarios e identificar métricas que permitan una posterior mejora			
OBSERVACIONES			
CRITERIO DE ACEPTACION	El usuario administrador accede a módulo de métricas		

## 3.2 Proceso de actualización Ionic v3 Ionic 5.4.16.

### 3.2.1 Estructura del proyecto

Uno de los principales cambios entre una aplicación en Ionic con la version 3 y la version 4 es la estructura y el diseño del proyecto, como podemos evidenciar con las siguiente comparativa.



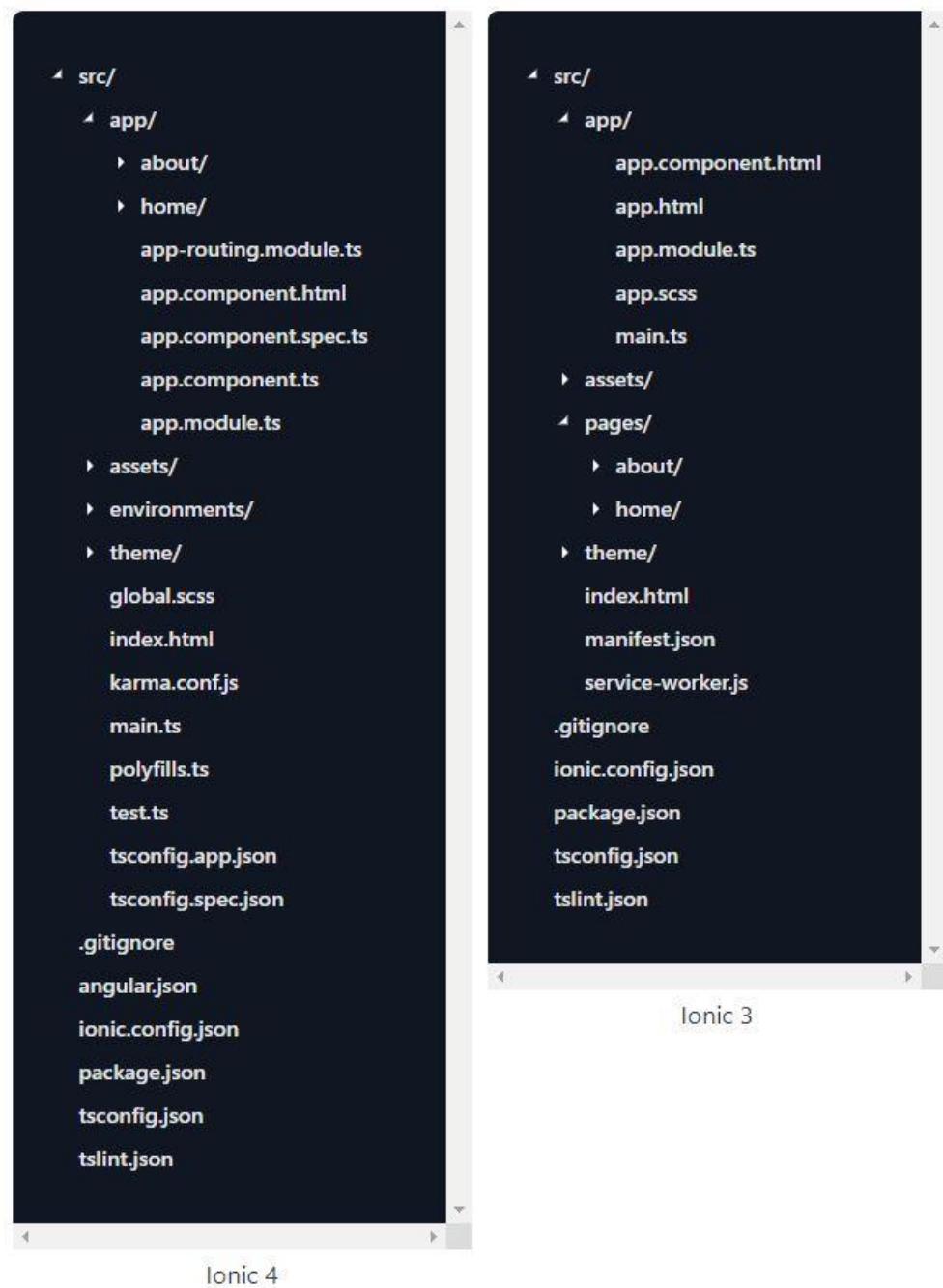


Ilustración 2.Comparativa estructura ionic v3-v4

La estructura que se implementó en la version en adelante hasta la version 5.4.16 que se provo actualmente a la fecha de elaboración de este documento. Es exactamente igual a lo que vendría a ser la estructura de un proyecto en angular

## Cambios de RxJS.

Entre V3 y V4, RxJS se actualizó a la versión 6. Esto cambia muchas de las rutas de importación de los operadores y las funciones básicas de RxJS. (ionic, 2021)

## Eventos de ciclo de vida.

Con V4, ahora podemos utilizar los eventos típicos proporcionados por Angular. Pero para ciertos casos, es posible que desee tener acceso a los eventos activados cuando un componente ha terminado de animarse durante su cambio de ruta. En este caso, la `ionViewWillEnter`, `ionViewDidEnter`, `ionViewWillLeave`, y `ionViewDidLeave` han portado de V3. Utilice estos eventos para coordinar acciones con el propio sistema de animaciones de Ionic.

Los eventos más antiguos como `ionViewDidLoad`, `ionViewCanLeave` y `ionViewCanEnter` han eliminado, y se deben usar las alternativas angulares adecuadas. (ionic, 2021)

## Componentes superpuestos.

En versiones anteriores de Ionic, los componentes de superposición como `Loading`, `Toast` o `Alert` se creaban de forma sincrónica. En Ionic v4, todos estos componentes se crean de forma asincrónica. Como resultado de esto, la API ahora se basa en promesas. (ionic, 2021)

## Navegación.

En V4, la navegación recibió la mayoría de los cambios. Ahora, en lugar de usar el propio Ionic `NavController`, lo integramos con el enrutador angular oficial. Esto no solo proporciona una experiencia de enrutamiento consistente entre aplicaciones, sino que es mucho más confiable. El equipo de Angular tiene una excelente guía en su sitio de documentos que cubre el enrutador con gran detalle.

Para proporcionar las animaciones específicas de la plataforma a las que los usuarios están acostumbrados, hemos creado `ion-router-outlet` para Angular Apps. Esto se comporta de manera similar a Angular, `router-outlet` pero proporciona una navegación basada en pilas (pestañas) y animaciones.

## Configuración de entorno de trabajo.

El primer paso para iniciar nuestro proyecto ionic consta de ajustar nuestro entorno de trabajo para lo cual necesitamos.

Instalar en nuestra maquina los siguientes programas que se pueden instalar tanto en Windows, linux y mac

- Git
- Nodejs
- Npm
- Cordova
- Ionic cli 3

Para poder hacer la actualización hay que estudiar los cambios en dependencias y requerimientos que cambian en el respectivo proyecto Ionic.

1. Instalar Nodejs: este nos permite acceder al administrador de paquetes npm el cual utilizaremos para descargar y crear nuestro proyecto. Para instalarlo: accedemos a <https://nodejs.org/es/>



Ilustración 3. Sitio nodejs.org/es

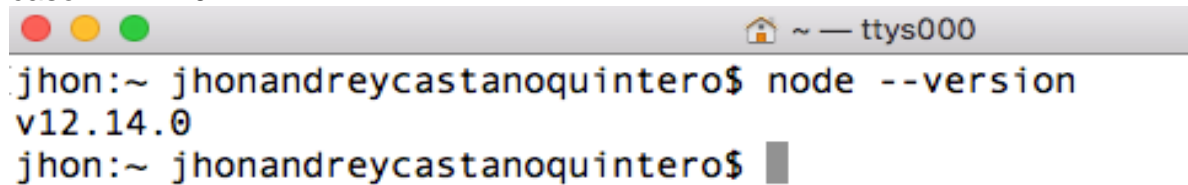
Al entrar en este sitio este nos identifica el sistema operativo para el caso de este proyecto estamos usando un sistema macOS de 64 bits. (recomendable instalar la versión LTS) la cual es una versión que se encuentra testada y tiene soporte durante más tiempo. Luego de descargar instalamos su ejecutable. Para sistemas operativos como Linux podemos utilizar en la terminal los comandos.

- sudo apt update
- sudo apt install Nodejs
- sudo apt install npm

**Nota:** con el fin de saber si nuestro ejecutable instaló correctamente nos dirigimos a CMD en Windows o el Terminal de comandos en Linux y Mac y ejecutamos el siguiente comando:

- `node --version`

Si la instalación es correcta nos devolverá la versión que instalamos para este caso v12.14.0



```

jhon:~ jhonandreycastanoquintero$ node --version
v12.14.0
jhon:~ jhonandreycastanoquintero$

```

Ilustración 4. Comando `node --version` mac

2. Luego de tener node js y su gestor de paquetes npm podremos instalar los paquetes necesarios para correr el cliente de ionic en nuestra máquina.

**Nota:** Este paso se utiliza para compilar con cordova la versión 3 de ionic a partir de la versión 4 se utiliza capacitor para lo cual podemos saltarnos este paso.

- `npm install -g cordova`

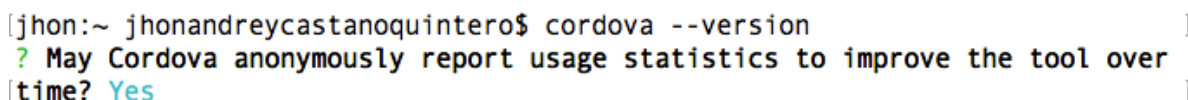
El flag -g nos indica que vamos a instalar nuestro paquete, para que pueda ser utilizado de forma global en nuestro sistema operativo.

**Nota:** en mac y Linux debemos incluir la palabra `sudo` para ya que esta instalación requiere permisos de administrador, este nos solicitara la contraseña de administrador para iniciar y continuar con la instalación.

- `sudo npm install -g cordova`

Para comprobar nuestra instalación de cordova escribimos el comando en nuestro terminal.

- `cordova --version`



```

jhon:~ jhonandreycastanoquintero$ cordova --version
? May Cordova anonymously report usage statistics to improve the tool over time? Yes

```

```

Thanks for opting into telemetry to help us improve cordova.
9.0.0 (cordova-lib@9.0.1)

```

Ilustración 5. Comando `cordova --version` mac

Cordova es el compilador que nos transforma a nuestra aplicación de ionic a un lenguaje que podamos utilizar con Android studio (aplicación para Android) o para Swift (aplicación ios).

En las versiones más recientes se puede compilar a través de capacitor, es la nueva tecnología a la fecha de redactado este documento que esta implementado ionic para compilar sus aplicaciones.

2. Procedemos a instalar ionic, al igual que cordova usamos el gestor de paquetes npm que nos suministra la instalación de Nodejs si queremos instalar una versión específica agregamos @ seguido de la versión.

- `npm install -g ionic@2.2.0`

Podemos consultar más información de cómo instalar ionic en su enlace oficial <https://ionicframework.com/docs/installation/cli> confirmamos instalación con el comando.

- `ionic --version`

```
jhon:~ jhonandreycastanoquintero$ ionic --version
5.4.13
```

3. Como la aplicación está elaborada con ionic 3 utilizando angular procedemos a instalar typescript con el gestor de paquetes.

- `npm install -g typescript`

Confirmamos que la instalación esta correcta con el comando

- `tsc --version`

```
jhon:~ jhonandreycastanoquintero$ tsc --version
Version 3.5.3
```

4. Una vez instalado ionic y Nodejs procedemos a instalar un editor de Código mi preferencia es VScode, aunque también se pueden utilizar editores como Atom, sublime, etc.

Para instalar VScode podemos descargarlo de <https://code.visualstudio.com/>

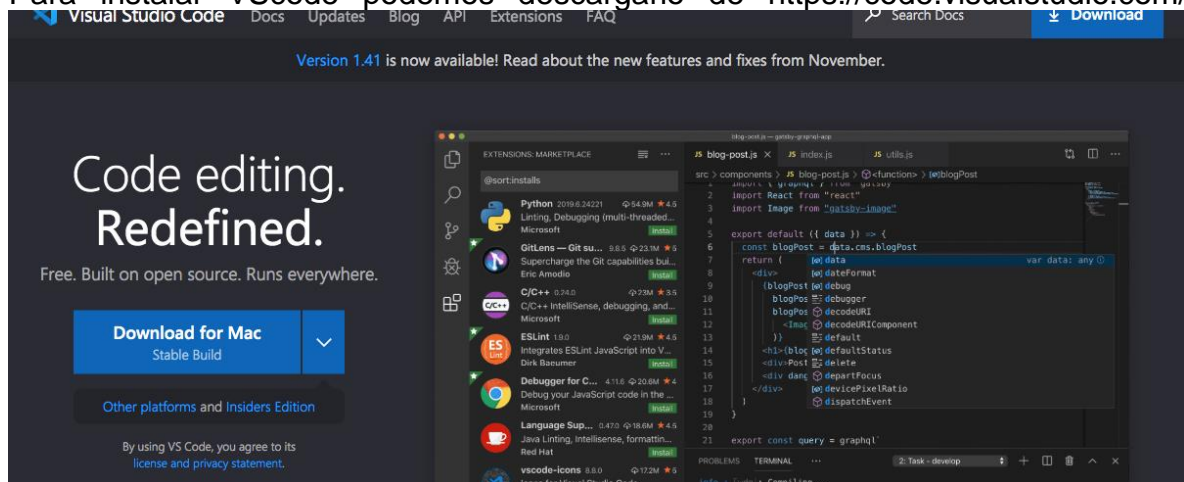


Ilustración 6. Sitio Download VScode

Al igual que con Nodejs descargamos y ejecutamos el archivo de instalación en Windows.

Una vez tenemos nuestro entorno de trabajo listo procedemos a la actualización de la app del framework Ionic v3 a v5 podemos seguir la siguiente guía en su sitio oficial <https://ionicframework.com/docs/building/migration>.

5. Para comenzar con la migración iniciamos un proyecto nuevo con el CLI de ionic para la cual posicionamos nuestra terminal en la carpeta que vamos a trabajar, luego de esto utilizamos el comando de ionic
  - ionic start liderazgo blank

A partir de la version 4 de ionic permite trabajar con frameworks de JavaScript diferentes a angular, como lo son react y vue. Pero como el proyecto que vamos a migrar sus componentes se encuentra en angular procedemos a instalar este lo seleccionamos, indicando al cli la estructura del proyecto en angular.

Pick a framework! 😊

Please select the JavaScript framework to use for your new app. To bypass this prompt next time, supply a value for the `--type` option.

```
? Framework: (Use arrow keys)
> Angular | https://angular.io
  React   | https://reactjs.org
```

---

Esperamos unos minutos mientras el CLI crea e instala nuestro entorno de trabajo Si todo sale bien nos saldría este mensaje

#### [INFO] Next Steps:

- Go to your newly created project: `cd ./liderazgo`
- Run `ionic serve` within the app directory to see your app
- Build features and components: <https://ion.link/scaffolding-docs>
- Run your app on a hardware or virtual device:  
<https://ion.link/running-docs>

6. Ahora vamos a clonar el proyecto al cual vamos a actualizar a la nueva version de ionic para lo cual necesitaremos git instalado en nuestro pc. Los sistemas operativos como linux y mac vienen con git incluido en nuestro terminal en caso de usar una dependencia sin git de Ubuntu lo podrías instalar desde la terminal utilizando el comando “sudo apt install git”, para instalar git en Windows lo descargamos del siguiente enlace <https://git-scm.com/download/win>

El proyecto para actualizar se encuentra en la version de ionic3 se encuentra disponible en el siguiente repositorio <https://github.com/nico0829/Proyecto-de-Grado.git>

Y la app actualizada para la version ionic5 se encuentra en el siguiente repositorio [https://github.com/jhancastano/proyecto\\_de\\_grado.git](https://github.com/jhancastano/proyecto_de_grado.git)

### 3.2.2 PROCESO DE AUTENTIFICACIÓN EN LA NUEVA VERSION DE IONIC.

Para el proceso de autenticación nos conectaremos mediante web services con la plataforma Moodle de la UTP, la cual tiene el siguiente enlace.

- <https://plataforma.utp.edu.co>

Podemos obtener el token de acceso de esta plataforma siempre y cuando estén habilitados por el administrador los webs services para esto podemos hacer la siguiente solicitud http tipo POST:

- <https://plataforma.utp.edu.co/login/token.php?username=USERNAME&password=PASSWORD&service=SERVICESHOSTNAME>

En nuestra petición POST las variables en mayúscula son aquellas que debemos reemplazar por la captura de datos.

- USERNAME
- PASSWORD
- SERVICESHOSTNAME

**Nota:** para obtener el token la variable SERVICESHOSTNAME debería remplazarse por **moodle\_mobile\_app**.

Si la petición sale exitosa nos enviaría una respuesta con el token de acceso similar a esta.

```
{
  "token": "9859148a89546f0efe716a58e340849b",
  "privatetoken": "8RpHJevJ42W7QN23OMkeYcdOYw3YfWgWGsak7WB3Z88wcApSCVZ9TgY6M5fE01m"
}
```

Este es el token que usaremos para identificar al usuario.

Luego solicitamos podemos traer los datos de perfil con la siguiente petición.

- [https://plataforma.utp.edu.co/webservice/rest/server.php?moodlewsrestformat=json&wsfunction=core\\_webservice\\_get\\_site\\_info&moodlewssettingfilter=true&moodlewssettingfileurl=true&wstoken=TOKEN](https://plataforma.utp.edu.co/webservice/rest/server.php?moodlewsrestformat=json&wsfunction=core_webservice_get_site_info&moodlewssettingfilter=true&moodlewssettingfileurl=true&wstoken=TOKEN)

En los anexos podremos encontrar los servicios que se pueden llamar para traer información de la plataforma Moodle.

Luego de tener la información de cómo obtener nuestro token procedemos a generar nuestros servicios y vistas en la aplicación de ionic5.

Para ello utilizamos el CLI de ionic con los siguientes comandos:

- ionic g page login

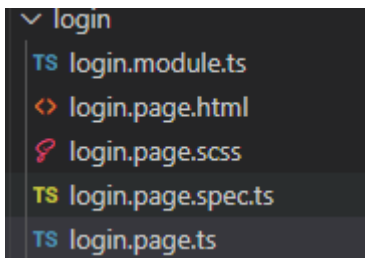


Ilustración 7.Resultado ionic g page login

- ionic g service services/authentication
- ionic g service services/authGuard

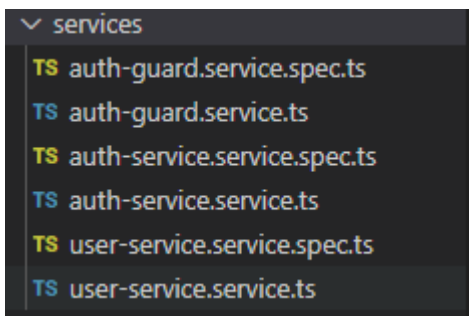


Ilustración 8.Resultado ionic g service

En nuestro archivo login.page.html ingresamos.



```

<ion-content>
  
  <form [formGroup]="loginForm">
    <ion-list>
      <ion-item>
        <ion-label position="stacked">Usuario</ion-label>
        <ion-input formControlName="username" type="text">
      </ion-item>

      <ion-item>
        <ion-label position="stacked">Contraseña</ion-label>
        <ion-input formControlName="password" type="text">
      </ion-item>
    </ion-list>

    <ion-row class="signin">
      <ion-col>
        <ion-button (click)="login()" color="primary" shape="full" expand="block">Ingresar</ion-button>
      </ion-col>
    </ion-row>
  </form>
</ion-content>

```

Ilustración 9.Template login moodle

Dando como resultado.

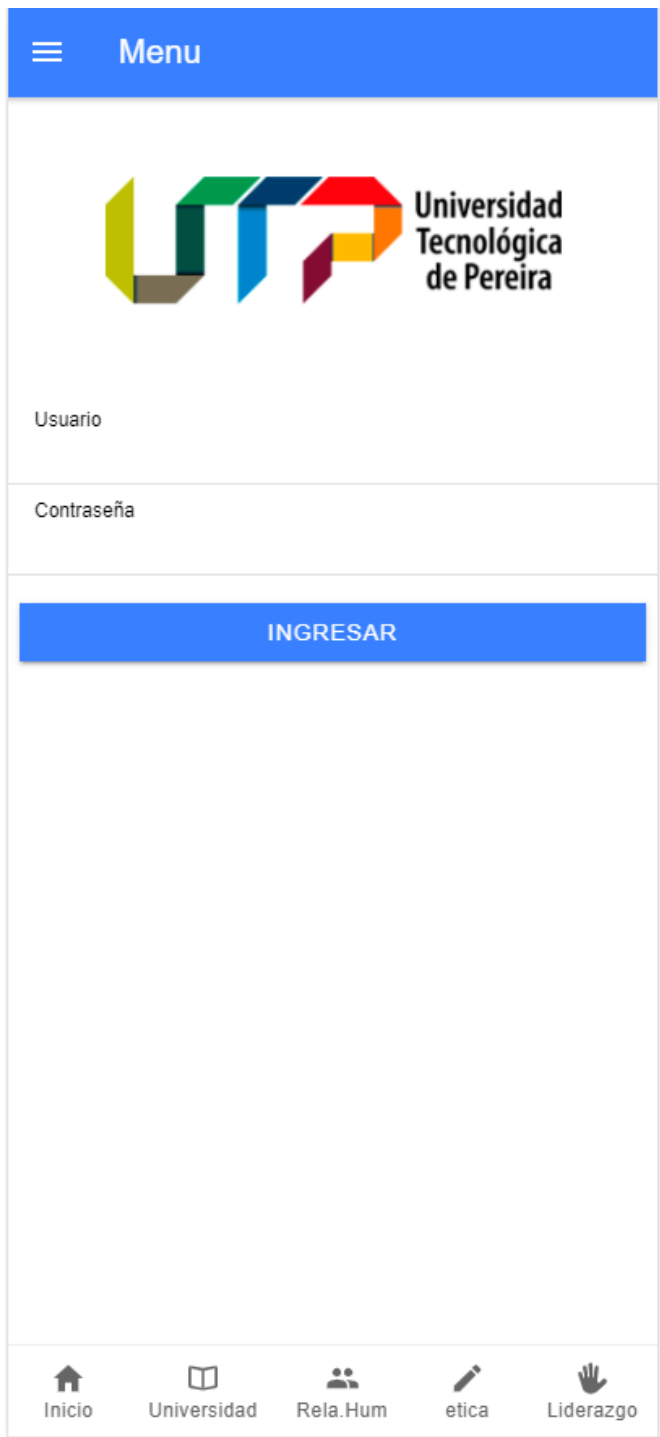


Ilustración 10.Vista Login

Para obtener el proyecto final lo podemos descargar del siguiente repositorio:  
[https://github.com/jhancastano/proyecto\\_de\\_grado](https://github.com/jhancastano/proyecto_de_grado)

El cual puede ser ejecutado con la imagen Docker que se encuentra dentro del repositorio como se explica a continuación.

```

1 FROM node:16-alpine
2
3 ENV IONIC_VERSION=5.4.16
4 ENV PORT=8200
5
6 WORKDIR /app
7 COPY package*.json ./
8
9 RUN npm install -g cordova ionic@${IONIC_VERSION}
10 RUN npm install
11
12 COPY . .
13
14 CMD ["ionic", "serve", "--port=${PORT}", "--address", "0.0.0.0"]

```

Ilustración 11.Imagen Docker proyecto

Podemos iniciar el proyecto en nuestro explorador instalando Docker y ejecutando los siguientes comandos:

- 1) Crear imagen.
  - docker build -t NAME\_CONTENEDOR PATH\_IMAGEN
- 2) Correr imagen.
  - docker run -it -p 8300:8100 NAME\_CONTENEDOR

## Capitulo IV: Aspectos administrativos

### 4.1 Recursos

item	costo	cantidad	total
computador	1000000	1	1000000
lapiceros	5000	1	5000
memoria usb 16gb	20000	1	20000
celular android	400000	1	400000
total			1425000

Ilustración 12.Recursos desarrollo

Los recursos fueron asumidos por el estudiante.

## 4.2 Cronograma

CRONOGRAMA																		
		Semanas																
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1. Análisis	1.1 Revisión de requisitos	x																
	1.2 Diseño de casos de uso		x															
	1.3 Diseño de diagrama de objetos, relaciones y jerarquía de contenidos			x														
	1.4 Diseño de diagrama de clases				x													
2. Diseño	2.1 Diseño de mockups				x													
	2.2 Diseño de diagrama de despliegue (modelo de configuración)					x												
	2.3 Diseño de diagrama de clases(diseño modelo vista controlador)						x											
	2.4 Diseño de modelo relación para la base de datos							x										
3. Construcción	3.1 Programación de casos de uso								x									
	3.2 Implementación base de datos									x								
4. Pruebas	4.1 Elaborar plan de pruebas										x							
	4.2 Pruebas unitarias											x						
	4.3 Pruebas integrales												x					
	4.4 Informe resultados de pruebas													x				
5. Implementación	5.1 Migrar base de datos															x		
	5.2 Instalar aplicación																x	
	5.3 Redactar manuales de usuario																	x

Ilustración 13. Cronograma proyecto

## Trabajo futuro

Completar los microservicios de manera escalable. De la mayoría de las funcionalidades de la plataforma Moodle.

Agregar componente que permita inscribir módulos por medio de códigos QR

## Conclusiones

- Es muy importante indicar las versiones y dependencias que se están usando al momento de desarrollar un software para configurar nuestro entorno de trabajo mucho más rápido.
- Los contenedores nos permiten evitar conflictos entre dependencias y versiones ya instaladas en nuestro sistema operativo
- Las aplicaciones desarrolladas con en base a frameworks de JavaScript nos permiten acelerar la velocidad de construcción de aplicaciones tanto para dispositivos móviles, como para web.
- Los microservicios nos permiten tener un método de comunicación entre diferentes plataformas para poder extender nuestros sistemas, permitiendo modularizar.

## Anexos

### Repositorio aplicativo

[https://github.com/jhancastano/proyecto\\_de\\_grado](https://github.com/jhancastano/proyecto_de_grado)

### Lista Web Services moodle\_mobile\_app

Nombre ws	Descripción
block_recentlyaccesseditems_get_recent_items	List of items a user has accessed most recently.
block_starredcourses_get_starred_courses	Get users starred courses.
core_badges_get_user_badges	Returns the list of badges awarded to a user.
core_block_get_course_blocks	Returns blocks information for a course.

core_block_get_dashboard_blocks	Returns blocks information for the given user dashboard.
core_block_fetch_addable_blocks	Returns all addable blocks in a given page.
core_blog_get_entries	Returns blog entries.
core_blog_view_entries	Trigger the blog_entries_viewed event.
core_calendar_create_calendar_events	Create calendar events
core_calendar_delete_calendar_events	Delete calendar events
core_calendar_get_action_events_by_course	Get calendar action events by course
core_calendar_get_action_events_by_courses	Get calendar action events by courses
core_calendar_get_action_events_by_tiemsort	Get calendar action events by tiemsort
core_calendar_get_allowed_event_types	Get the type of events a user can create in the given course.
core_calendar_get_calendar_access_information	Convenience function to retrieve some permissions/access information for the given course calendar.
core_calendar_get_calendar_day_view	Fetch the day view data for a calendar
core_calendar_get_calendar_event_by_id	Get calendar event by id
core_calendar_get_calendar_events	Get calendar events
core_calendar_get_calendar_monthly_view	Fetch the monthly view data for a calendar

core_calendar_get_calendar_upcoming_view	Fetch the upcoming view data for a calendar
core_calendar_submit_create_update_form	Submit form data for event form
core_calendar_update_event_start_day	Update the start day (but not time) for an event.
core_calendar_get_calendar_export_token	Return the auth token required for exporting a calendar.
core_comment_add_comments	Adds a comment or comments.
core_comment_delete_comments	Deletes a comment or comments.
core_comment_get_comments	Return all the comments for an specific component, context.
core_competency_competency_viewed	Report a competency as being viewed
core_competency_delete_evidence	Delete an evidence
core_competency_get_scale_values	Get the values for a scale
core_competency_grade_competency_in_course	Rate a user's competency in a course
core_competency_list_course_competencies	List the competencies in a course
core_competency_user_competency_plan_viewed	Report the archive of user's competency as being viewed
core_competency_user_competency_viewed	Report a user's competency as being viewed

core_competency_user_competency_viewed_in_course	Report a user's competency as being viewed from a course
core_competency_user_competency_viewed_in_plan	Report a user's competency as being viewed from a learning plan
core_completion_get_activities_completion_status	Return the activities completion status for a user in a course.
core_completion_get_course_completion_status	Returns course completion status.
core_completion_mark_course_self_completed	Update the course completion status for the current user (if course self-completion is enabled).
core_completion_update_activity_completion_status_manually	Update completion status for the current user in an activity, only for activities with manual tracking.
core_course_check_updates	Check if there is updates affecting the user for the given course and contexts.
core_course_get_categories	get categories
core_course_get_contents	get course content (modules + web service file urls)
core_course_get_course_module	Return information about a course module.



core_course_get_course_module_by_instance	Return information about a given module name and instance id.
core_course_get_courses	Return course details
core_course_get_courses_by_field	Get courses matching a specific field (id/s, shortname, idnumber, category)
core_course_get_enrolled_courses_by_timeline_classification	List of enrolled courses for the given timeline classification (past, inprogress, or future).
core_course_get_recent_courses	List of courses a user has accessed most recently.
core_course_get_updates_since	Check if there are updates affecting the user for the given course since the given time stamp.
core_course_get_user_administration_options	Return a list of administration options in a set of courses that are available or not for the current user.
core_course_get_user_navigation_options	Return a list of navigation options in a set of courses that are available or not for the current user.
core_course_search_courses	Search courses by (name, module, block, tag).

core_course_set_favourite_courses	Add a list of courses to the list of favourite courses.
core_course_view_course	Log that the course was viewed
core_enrol_get_course_enrolment_methods	Get the list of course enrolment methods
core_enrol_get_enrolled_users	Get enrolled users by course id
core_enrol_get_users_courses	get list of course ids that a user is enrolled in (if you are allowed to see that)
core_grades_grader_gradingpanel_point_fetch	Fetch the data required to display the grader grading panel for simple grading, creating the grade item if required
core_grades_grader_gradingpanel_point_store	Store the data required to display the grader grading panel for simple grading
core_grades_grader_gradingpanel_scale_fetch	Fetch the data required to display the grader grading panel for scale-based grading, creating the grade item if required
core_grades_grader_gradingpanel_scale_store	Store the data required to display the grader grading panel for scale-based grading

core_group_get_activity_allowed_groups	Gets a list of groups that the user is allowed to access within the specified activity.
core_group_get_activity_groupmode	Returns effective groupmode used in a given activity.
core_group_get_course_groupings	get all groupings in specified course
core_group_get_course_groups	Returns all groups in specified course
core_group_get_course_user_groups	Returns all groups in specified course for the specified user.
core_h5p_get_trusted_h5p_file	Get the H5P file cleaned for Mobile App.
core_message_block_contacts	Block contacts
core_message_block_user	Blocks a user
core_message_confirm_contact_request	Confirms a contact request
core_message_create_contact_request	Creates a contact request
core_message_create_contacts	Add contacts to the contact list
core_message_data_for_messagearea_contacts	Retrieve the template data for the contact list
core_message_data_for_messagearea_conversations	Retrieve the template data for the conversation list
core_message_data_for_messagearea_messages	Retrieve the template data for the messages

core_message_data_for_messagearea_search_messages	Retrieve the template data for searching for messages
core_message_decline_contact_request	Declines a contact request
core_message_delete_contacts	Remove contacts from the contact list
core_message_delete_conversation	Deletes a conversation.
core_message_delete_conversations_by_id	Deletes a list of conversations.
core_message_delete_message	Deletes a message.
core_message_delete_message_for_all_users	Deletes a message for all users.
core_message_get_blocked_users	Retrieve a list of users blocked
core_message_get_contact_requests	Returns contact requests for a user
core_message_get_contacts	Retrieve the contact list
core_message_get_conversation	Retrieve a conversation for a user
core_message_get_conversation_between_users	Retrieve a conversation for a user between another user
core_message_get_conversation_counts	Retrieve a list of conversation counts, indexed by type.
core_message_get_conversation_members	Retrieve a list of members in a conversation
core_message_get_conversation_messages	Retrieve the conversation messages and relevant member information

core_message_get_conversations	Retrieve a list of conversations for a user
core_message_get_member_info	Retrieve a user message profiles
core_message_get_messages	Retrieve a list of messages sent and received by a user (conversations, notifications or both)
core_message_get_received_contact_requests_count	Gets the number of received contact requests
core_message_get_self_conversation	Retrieve a self-conversation for a user
core_message_get_unread_conversations_count	Retrieve the count of unread conversations for a given user
core_message_get_user_contacts	Retrieve the contact list
core_message_get_user_message_preferences	Get the message preferences for a given user.
core_message_get_user_notification_preferences	Get the notification preferences for a given user.
core_message_mark_all_conversation_messages_as_read	Mark all conversation messages as read for a given user
core_message_mark_all_messages_as_read	Mark all messages as read for a given user
core_message_mark_all_notifications_as_read	Mark all notifications as read for a given user

core_message_mark_message_read	Mark a single message as read, trigger message_viewed event.
core_message_mark_notification_read	Mark a single notification as read, trigger notification_viewed event.
core_message_message_processor_config_form	Process the message processor config form
core_message_message_search_users	Retrieve the data for searching for people
core_message_mute_conversations	Mutes a list of conversations
core_message_search_contacts	Search for contacts
core_message_send_instant_messages	Send instant messages
core_message_send_messages_to_conversation	Send messages to an existing conversation between users
core_message_set_favourite_conversations	Mark a conversation or group of conversations as favourites/starred conversations.
core_message_unblock_contacts	Unblock contacts
core_message_unblock_user	Unblocks a user
core_message_unmute_conversations	Unmutes a list of conversations

core_message_unset_favourite_conversations	Unset a conversation or group of conversations as favourites/starred conversations.
core_notes_create_notes	Create notes
core_notes_delete_notes	Delete Notes
core_notes_get_course_notes	Returns all notes in specified course (or site) for the specified user.
core_notes_view_notes	Simulates the web interface view of notes/index.php: trigger events.
core_question_update_flag	Update the flag state of a question attempt.
core_rating_add_rating	Rates an item.
core_rating_get_item_ratings	Retrieve all the ratings for an item.
core_table_get_dynamic_table_content	Get the dynamic table content raw html
core_tag_get_tag_areas	Retrieves existing tag areas.
core_tag_get_tag_cloud	Retrieves a tag cloud for the given collection and/or query search.
core_tag_get_tag_collections	Retrieves existing tag collections.
core_tag_get_tagindex	Gets tag index page for one tag and one tag area

core_tag_get_tagindex_per_area	Gets tag index page per different areas.
core_user_add_user_device	Store mobile user devices information for PUSH Notifications.
core_user_add_user_private_files	Copy files from a draft area to users private files area.
core_user_agree_site_policy	Agree the site policy for the current user.
core_user_get_course_user_profiles	Get course user profiles by id
core_user_get_private_files_info	Returns general information about files in the user private files area.
core_user_get_user_preferences	Return user preferences.
core_user_get_users_by_field	Retrieve users information for a specified unique field - If you want to do a user search, use core_user_get_users()
core_user_remove_user_device	Remove a user device from the Moodle database.
core_user_set_user_preferences	Set user preferences.
core_user_update_picture	Update or delete the user picture in the site
core_user_update_user_preferences	Update a user's preferences



core_user_view_user_list	Simulates the web-interface view of user/index.php (triggering events).
core_user_view_user_profile	Simulates the web-interface view of user/view.php and user/profile.php (triggering events).
core_webservice_get_site_info	Return some site info / user info / list web service functions
core_xapi_statement_post	Post an xAPI statement.
enrol_guest_get_instance_info	Return guest enrolment instance information.
enrol_self_enrol_user	Self enrol the current user in the given course.
enrol_self_get_instance_info	self enrolment instance information.
gradereport_overview_get_course_grades	Get the given user courses final grades
gradereport_overview_view_grade_report	Trigger the report view event
gradereport_user_get_grade_items	Returns the complete list of grade items for users in a course
gradereport_user_get_grades_table	Get the user/s report grades table for a course

gradereport_user_view_grade_report	Trigger the grade_report_viewed event when a user view his grade report (this web service will be initially used by the mobile app so information can be logged in the Moodle side when a user see the same page in the app).
message_airnotifier_are_notification_preferences_configured	Check if the users have notification preferences configured yet
message_airnotifier_enable_device	Enables or disables a registered user device so it can receive Push notifications
message_airnotifier_get_user_devices	Return the list of mobile devices that are registered in Moodle for the given user
message_airnotifier_is_system_configured	Check whether the airnotifier settings have been configured
mod_assign_lock_submissions	Prevent students from making changes to a list of submissions
mod_assign_reveal_identities	Reveal the identities for a blind marking assignment
mod_assign_revert_submissions_to_draft	Reverts the list of submissions to draft status
mod_assign_save_grade	Save a grade update for a single student.

mod_assign_save_grades	Save grade updates for one or more students.
mod_assign_save_submission	Update the current students submission
mod_assign_save_user_extensions	Save a list of assignment extensions
mod_assign_set_user_flags	Sets the specified user_flags for an assignment
mod_assign_submit_for_grading	Submit the current students assignment for grading
mod_assign_submit_grading_form	Submit the grading form data via ajax
mod_assign_unlock_submissions	Allow students to make changes to a list of submissions
mod_assign_view_assign	Update the module completion status.
mod_assign_view_grading_table	Trigger the grading_table_viewed event.
mod_assign_view_submission_status	Trigger the submission status viewed event.
mod_book_get_books_by_courses	Returns a list of book instances in a provided set of courses.
mod_book_view_book	Trigger the course module viewed event and update the module completion status.

mod_chat_get_chat_latest_messages	Get the latest messages from the given chat session.
mod_chat_get_chat_users	Get the list of users in the given chat session.
mod_chat_get_chats_by_courses	Returns a list of chat instances in a provided set of courses.
mod_chat_get_session_messages	Retrieves messages of the given chat session.
mod_chat_get_sessions	Retrieves chat sessions for a given chat.
mod_chat_login_user	Log a user into a chat room in the given chat.
mod_chat_send_chat_message	Send a message on the given chat session.
mod_chat_view_chat	Trigger the course module viewed event and update the module completion status.
mod_choice_delete_choice_responses	Delete the given submitted responses in a choice.
mod_choice_get_choice_options	Retrieve options for a specific choice.
mod_choice_get_choice_results	Retrieve users results for a given choice.
mod_choice_get_choices_by_courses	Returns a list of choice instances in a provided set of courses.
mod_choice_submit_choice_response	Submit responses to a specific choice item.

mod_choice_view_choice	Trigger the course module viewed event and update the module completion status..
mod_data_get_entries	Return the complete list of entries of the given database.
mod_data_get_entry	Return one entry record from the database, including contents optionally.
mod_data_get_fields	Return the list of configured fields for the given database.
mod_data_search_entries	Search for entries in the given database.
mod_data_update_entry	Updates an existing entry.
mod_data_view_database	Simulate the view.php web interface data: trigger events, completion, etc...
mod_feedback_get_analysis	Retrieves the feedback analysis.
mod_feedback_get_current_completed_tmp	Returns the temporary completion record for the current user.
mod_feedback_get_feedback_access_information	Return access information for a given feedback.

mod_feedback_get_feedbacks_by_courses	Returns a list of feedbacks in a provided list of courses, if no list is provided all feedbacks that the user can view will be returned.
mod_feedback_get_finished_responses	Retrieves responses from the last finished attempt.
mod_feedback_get_items	Returns the items (questions) in the given feedback.
mod_feedback_get_last_completed	Retrieves the last completion record for the current user.
mod_feedback_get_non_respondents	Retrieves a list of students who didn't submit the feedback.
mod_feedback_get_page_items	Get a single feedback page items.
mod_feedback_get_responses_analysis	Return the feedback user responses analysis.
mod_feedback_get_unfinished_responses	Retrieves responses from the current unfinished attempt.
mod_feedback_launch_feedback	Starts or continues a feedback submission.
mod_feedback_process_page	Process a jump between pages.
mod_feedback_view_feedback	Trigger the course module viewed event and update the module completion status.

mod_folder_get_folders_by_courses	Returns a list of folders in a provided list of courses, if no list is provided all folders that the user can view will be returned. Please note that this WS is not returning the folder contents.
mod_folder_view_folder	Trigger the course module viewed event and update the module completion status.
mod_forum_add_discussion	Add a new discussion into an existing forum.
mod_forum_add_discussion_post	Create new posts into an existing discussion.
mod_forum_can_add_discussion	Check if the current user can add discussions in the given forum (and optionally for the given group).
mod_forum_delete_post	Deletes a post or a discussion completely when the post is the discussion topic.
mod_forum_get_discussion_post	Get a particular discussion post.
mod_forum_get_discussion_posts	Returns a list of forum posts for a discussion.
mod_forum_get_forum_access_information	Return capabilities information for a given forum.

mod_forum_get_forum_discussion_posts	Returns a list of forum posts for a discussion.
mod_forum_get_forum_discussions	Returns a list of forum discussions optionally sorted and paginated.
mod_forum_get_forum_discussions_paginated	Returns a list of forum discussions optionally sorted and paginated.
mod_forum_get_forums_by_courses	Returns a list of forum instances in a provided set of courses, if no courses are provided then all the forum instances the user has access to will be returned.
mod_forum_prepare_draft_area_for_post	Prepares a draft area for editing a post.
mod_forum_set_lock_state	Set the lock state for the discussion
mod_forum_set_pin_state	Set the pin state
mod_forum_set_subscription_state	Set the subscription state
mod_forum_toggle_favourite_state	Toggle the favourite state
mod_forum_update_discussion_post	Updates a post or a discussion topic post.



mod_forum_view_forum	Trigger the course_module_viewed event when a user view a forum (this web service will be initially used by the mobile app so information can be logged in the Moodle side when a user see the same page in the app).
mod_forum_view_forum_discussion	Trigger the discussion_viewed event when a user view a forum discussion (this web service will be initially used by the mobile app so information can be logged in the Moodle side when a user see the same page in the app).
mod_glossary_add_entry	Add a new entry to a given glossary
mod_glossary_get_entries_by_search	Get entries matching a search query
mod_glossary_get_entries_by_term	Get entries using term filtering
mod_glossary_get_entries_to_approve	Get entries to be approved
mod_glossary_get_entry_by_id	Get an entry by ID
mod_glossary_get_glossaries_by_courses	Get the glossaries in courses
mod_glossary_view_entry	Notify that a glossary entry was viewed

mod_glossary_view_glossary	Notify that a glossary was viewed
mod_imsdp_get_imsdps_by_courses	Returns a list of imscp instances in a provided set of courses.
mod_imsdp_view_imsdp	Trigger the course module viewed event and update the module completion status.
mod_lesson_get_lesson	Return information of a given lesson.
mod_lesson_get_lesson_access_information	Return access information for a given lesson.
mod_lesson_get_lessons_by_courses	Returns a list of lessons in a provided list of courses, if no list is provided all lessons that the user can view will be returned.
mod_lesson_get_page_data	Return information of a given page, including its contents.
mod_lesson_get_pages	Return the list of pages in a lesson (based on the user permissions).
mod_lesson_get_pages_possible_jumps	Return all the possible jumps for the pages in a given lesson.
mod_lesson_get_questions_attempts	Return the list of questions attempts in a given lesson.

mod_lesson_get_user_attempt	Return information about the given user attempt (including answers).
mod_lesson_get_user_attempt_grade	Return grade information in the attempt for a given user.
mod_lesson_get_user_grade	Return the final grade in the lesson for the given user.
mod_lesson_get_user_timers	Return the timers in the current lesson for the given user.
mod_lesson_launch_attempt	Starts a new attempt or continues an existing one.
mod_lesson_process_page	Processes page responses.
mod_lesson_view_lesson	Trigger the course module viewed event and update the module completion status.
mod_lti_get_lti_by_courses	Returns a list of lti instances in a provided set of courses.
mod_lti_get_tool_launch_data	Return the launch data for a given external tool.
mod_lti_view_lti	Trigger the course module viewed event and update the module completion status.

mod_page_get_pages_by_courses	Returns a list of pages in a provided list of courses, if no list is provided all pages that the user can view will be returned.
mod_page_view_page	Trigger the course module viewed event and update the module completion status.
mod_quiz_get_attempt_access_information	Return access information for a given attempt in a quiz.
mod_quiz_get_attempt_data	Returns information for the given attempt page for a quiz attempt in progress.
mod_quiz_get_attempt_review	Returns review information for the given finished attempt, can be used by users or teachers.
mod_quiz_get_attempt_summary	Returns a summary of a quiz attempt before it is submitted.
mod_quiz_get_combined_review_options	Combines the review options from a number of different quiz attempts.
mod_quiz_get_quiz_access_information	Return access information for a given quiz.

mod_quiz_get_quiz_feedback_for_grade	Get the feedback text that should be show to a student who got the given grade in the given quiz.
mod_quiz_get_quiz_required_qtypes	Return the potential question types that would be required for a given quiz.
mod_quiz_get_quizzes_by_courses	Returns a list of quizzes in a provided list of courses, if no list is provided all quizzes that the user can view will be returned.
mod_quiz_get_user_attempts	Return a list of attempts for the given quiz and user.
mod_quiz_get_user_best_grade	Get the best current grade for the given user on a quiz.
mod_quiz_process_attempt	Process responses during an attempt at a quiz and also deals with attempts finishing.
mod_quiz_save_attempt	Processes save requests during the quiz.This function is intended for the quiz auto-save feature.
mod_quiz_start_attempt	Starts a new attempt at a quiz.
mod_quiz_view_attempt	Trigger the attempt viewed event.

mod_quiz_view_attempt_review	Trigger the attempt reviewed event.
mod_quiz_view_attempt_summary	Trigger the attempt summary viewed event.
mod_quiz_view_quiz	Trigger the course module viewed event and update the module completion status.
mod_survey_view_survey	Trigger the course module viewed event and update the module completion status.
mod_url_get_urls_by_courses	Returns a list of urls in a provided list of courses, if no list is provided all urls that the user can view will be returned.
mod_url_view_url	Trigger the course module viewed event and update the module completion status.
mod_wiki_edit_page	Save the contents of a page.
mod_wiki_get_page_contents	Returns the contents of a page.
mod_wiki_get_page_for_editing	Locks and retrieves info of page-section to be edited.
mod_wiki_get_subwiki_files	Returns the list of files for a specific subwiki.
mod_wiki_get_subwiki_pages	Returns the list of pages for a specific subwiki.

mod_wiki_get_subwikis	Returns the list of subwikis the user can see in a specific wiki.
mod_wiki_get_wikis_by_courses	Returns a list of wiki instances in a provided set of courses, if no courses are provided then all the wiki instances the user has access to will be returned.
mod_workshop_get_assessment	Retrieves the given assessment.
mod_workshop_get_assessment_form_definition	Retrieves the assessment form definition.
mod_workshop_get_grades	Returns the assessment and submission grade for the given user.
mod_workshop_get_grades_report	Retrieves the assessment grades report.
mod_workshop_get_reviewer_assessments	Retrieves all the assessments reviewed by the given user.
mod_workshop_get_submission	Retrieves the given submission.
mod_workshop_get_submission_assessments	Retrieves all the assessments of the given submission.

(moodle, 2021)

## Bibliografía

*Ajax*. (s.f.). Recuperado el 10 de 6 de 2021, de Wikipedia, la enciclopedia libre:  
<http://es.wikipedia.org/wiki/AJAX>

*AngularJS*. (s.f.). Recuperado el 10 de 6 de 2021, de Wikipedia, la enciclopedia libre: <http://es.wikipedia.org/wiki/AngularJS>

*Apache Cordova*. (s.f.). Recuperado el 10 de 6 de 2021, de Wikipedia, la enciclopedia libre: <http://es.wikipedia.org/wiki/PhoneGap>

*Arquitectura de microservicios*. (s.f.). Recuperado el 10 de 6 de 2021, de Wikipedia, la enciclopedia libre: [http://es.wikipedia.org/wiki/Arquitectura\\_de\\_microservicios](http://es.wikipedia.org/wiki/Arquitectura_de_microservicios)

*Componentes web*. (s.f.). Recuperado el 10 de 6 de 2021, de Wikipedia, la enciclopedia libre: [http://es.wikipedia.org/wiki/Componentes\\_web](http://es.wikipedia.org/wiki/Componentes_web)

Cruz, J. F. (1998). *Acerca de la programación en el shell de UNIX*. Recuperado el 10 de 6 de 2021, de [http://red-mat.unam.mx/foro/volumenes/vol007/prog\\_unx.pdf](http://red-mat.unam.mx/foro/volumenes/vol007/prog_unx.pdf)

*Docker*. (s.f.). Recuperado el 10 de 6 de 2021, de Wikipedia, la enciclopedia libre: [http://es.wikipedia.org/wiki/Docker\\_\(software\)](http://es.wikipedia.org/wiki/Docker_(software))

*Espacio de nombres XML*. (s.f.). Recuperado el 10 de 6 de 2021, de Wikipedia, la enciclopedia libre: [http://es.wikipedia.org/wiki/Espacio\\_de\\_nombres\\_XML](http://es.wikipedia.org/wiki/Espacio_de_nombres_XML)

*Experiencia de usuario*. (s.f.). Recuperado el 10 de 6 de 2021, de Wikipedia, la enciclopedia libre: [http://es.wikipedia.org/wiki/Experiencia\\_de\\_usuario](http://es.wikipedia.org/wiki/Experiencia_de_usuario)

*Framework para aplicaciones web*. (s.f.). Recuperado el 10 de 6 de 2021, de Wikipedia, la enciclopedia libre: [http://es.wikipedia.org/wiki/Framework\\_para\\_aplicaciones\\_web](http://es.wikipedia.org/wiki/Framework_para_aplicaciones_web)

hat, R. (10 de 6 de 2021). *www.redhat.com*. Obtenido de [www.redhat.com](https://www.redhat.com/es/topics/api/what-is-a-rest-api): <https://www.redhat.com/es/topics/api/what-is-a-rest-api>

*Hoja de estilos en cascada*. (s.f.). Recuperado el 10 de 6 de 2021, de Wikipedia, la enciclopedia libre: [http://es.wikipedia.org/wiki/Hoja\\_de\\_estilos\\_en\\_cascada](http://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada)

*HTML*. (s.f.). Recuperado el 10 de 6 de 2021, de Wikipedia, la enciclopedia libre: <http://es.wikipedia.org/wiki/HTML>

*Interfaz gráfica de usuario*. (s.f.). Recuperado el 10 de 6 de 2021, de Wikipedia, la enciclopedia libre: [http://es.wikipedia.org/wiki/Interfaz\\_gráfica\\_de\\_usuario](http://es.wikipedia.org/wiki/Interfaz_gráfica_de_usuario)

ionic. (11 de 06 de 2021). *ionic migration*. Obtenido de ionic: <https://ionicframework.com/docs/reference/migration>

*Javascript*. (s.f.). Recuperado el 10 de 6 de 2021, de Wikipedia, la enciclopedia libre: <http://es.wikipedia.org/wiki/JavaScript>

*JSON*. (s.f.). Recuperado el 10 de 6 de 2021, de Wikipedia, la enciclopedia libre: <http://es.wikipedia.org/wiki/JSON>

*JSON Web Token*. (s.f.). Recuperado el 12 de 6 de 2021, de Wikipedia, la enciclopedia libre: [http://es.wikipedia.org/wiki/JSON\\_Web\\_Token](http://es.wikipedia.org/wiki/JSON_Web_Token)

*Lazy loading*. (s.f.). Recuperado el 14 de 6 de 2021, de Wikipedia, la enciclopedia libre: [http://es.wikipedia.org/wiki/Lazy\\_loading](http://es.wikipedia.org/wiki/Lazy_loading)

*Moodle*. (s.f.). Recuperado el 10 de 6 de 2021, de Wikipedia, la enciclopedia libre: <http://es.wikipedia.org/wiki/Moodle>

moodle. (14 de 06 de 2021). *moodle*. Obtenido de moodle: [https://docs.moodle.org/dev/Web\\_service\\_API\\_functions](https://docs.moodle.org/dev/Web_service_API_functions)

*Node.js*. (s.f.). Recuperado el 10 de 6 de 2021, de Wikipedia, la enciclopedia libre: <http://es.wikipedia.org/wiki/Node.js>

*Npm*. (s.f.). Recuperado el 10 de 6 de 2021, de Wikipedia, la enciclopedia libre: <http://es.wikipedia.org/wiki/Npm>

*OAuth*. (s.f.). Recuperado el 12 de 6 de 2021, de Wikipedia, la enciclopedia libre: <http://es.wikipedia.org/wiki/OAuth>



*Progressive Web Apps*. (s.f.). Recuperado el 10 de 6 de 2021, de Wikipedia, la enciclopedia libre: [http://es.wikipedia.org/wiki/Progressive\\_Web\\_Apps](http://es.wikipedia.org/wiki/Progressive_Web_Apps)

*Servicio web*. (s.f.). Recuperado el 10 de 6 de 2021, de Wikipedia, la enciclopedia libre: [http://es.wikipedia.org/wiki/Servicio\\_web](http://es.wikipedia.org/wiki/Servicio_web)

*Shell de Unix*. (s.f.). Recuperado el 10 de 6 de 2021, de Wikipedia, la enciclopedia libre: [http://es.wikipedia.org/wiki/Shell\\_de\\_Unix](http://es.wikipedia.org/wiki/Shell_de_Unix)

*TypeScript*. (s.f.). Recuperado el 10 de 6 de 2021, de Wikipedia, la enciclopedia libre: <http://es.wikipedia.org/wiki/TypeScript>

*Visual Studio Code*. (s.f.). Recuperado el 10 de 6 de 2021, de Wikipedia, la enciclopedia libre: [http://es.wikipedia.org/wiki/Visual\\_Studio\\_Code](http://es.wikipedia.org/wiki/Visual_Studio_Code)

*XML-RPC*. (s.f.). Recuperado el 10 de 6 de 2021, de Wikipedia, la enciclopedia libre: <http://es.wikipedia.org/wiki/XML-RPC>